IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 09/887,524
Filing Date 06/21/2001
Inventorship..Simon et al.
Applicant... Microsoft Corporation
Group Art Unit 2162
Examiner ....Truoung, Cam Y T
Attorney's Docket No. MS1-744US
Title: *Automated Generator of Input-Validation Filters*

CoverPage for the Amended Appeal Brief

The Office sent a "Notification of Non-Compliant Appeal Brief (37 CFR 41.37)" on 1/5/2006. In response to "incorrect copy of the appealed claims" (item #7 on the numbered list of possible non-compliant elements), Applicant submits this Amended Appeal Brief with the corrected claim set herein.

In that Notification, the Office also indicated that the Appeal Brief did not contain a "concise explanation of the subject-matter defined in each of the independent claims" (item #4) and a "concise statement of each ground of rejection" (item #5). On 1/23/2006, Examiner Truoung was very kind and discussed this matter briefly with Kasey Christie (the undersigned attorney for the Applicant). In the conversation, the Examiner concluded that the original unamended Appeal Brief satisfied items #4 and #5 referenced above.

Respectfully Submitted,

Dated: _1·23 06_   By: _____

Kasey C. Christie
Reg. No. 40559
(509) 324-9256 x232
kasey@leehayes.com
www.leehayes.com

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 09/887,524
Filing Date 06/21/2001
Inventorship ....Simon et al.
Applicant.. Microsoft Corporation
Group Art Unit 2162
Examiner ....Truoung, Cam Y T
Attorney's Docket No. MS1-744US
Title: *Automated Generator of Input-Validation Filters*

AMENDED APPEAL BRIEF

In response to Notification of Non-Compliant Appeal Brief (37 CFR 41.37) dated 1/5/2006

To: MS: Appeal Brief - Patents
        Commissioner for Patents
        P.O. Box 1450
        Alexandria, VA 22313-1450

From: Kasey C. Christie   (Tel. 509-324-9256; Fax 509-323-8979)
        Customer No. 22801

Pursuant to 37 C.F.R. §1.192, Applicant hereby submits an appeal brief for Application No. 09/887,524. A Notice of Appeal was filed August 11, 2005. Accordingly, Applicant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com

TABLE OF CONTENTS

(1)   Real Party in Interest

The real party in interest is the Microsoft Corporation, the assignee of all right and title to the subject invention.

(2)   Related Appeals, Interferences, and Judicial Proceedings

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

Claims 1-21, 24-33, 42, 45-51, 54-58 stand rejected and are pending in this Application, and are set forth in the Appendix of Appealed Claims on page 30. Claims 1-58 were originally filed in the Application.   No claims have been allowed. Claims 22, 23, 34-41, 43, 44, 52, 53 have been canceled, withdrawn, and/or non-elected. Claims 1, 4, 19, 42, and 50 have been amended.

Claims 1-21, 24-33, 42, 45-51, 54-58 are subject of this appeal and stand rejected as set forth in a Final Office Action dated March 11, 2005 (hereinafter, the "FINAL ACTION").

As set forth in the FINAL ACTION, claims 1-21, 24-33, 42, 45-51, 54-58 stand rejected under USC § 103(a) as being obvious in light of a combination of two or more of the following references:

- **Fields**: *Fields et al.*, US Patent No. 6,605,120 (issued Aug. 12, 2003);

- **Lynch**: *Lynch et al.*, US Patent No. 6,558,431 (issued May 6, 2003);

- **Motoyama**: *Motoyama et al.*, US Patent No. 6,085,196 (issued June 4, 2000).

(4)   Status of Amendments

The Applicant responded to a non-final Office Action issued on August 27, 2004 (hereinafter, the "NON-FINAL ACTION"). In that response, Applicant canceled claims 22, 23, 34-41, 43, 44, 52, 53 and amended claims 1, 4, 19, 42, and 50. Applicant traversed all substantive rejections.

After that, the FINAL ACTION issued on March 11, 2005—the action dismissing Applicant's traversal and maintaining the rejection of all pending claims. In Applicant's response to the FINAL ACTION, Applicant traversed all substantive rejections, amended no other claims, and canceled no other claims. No other amendments have been filed subsequent to the FINAL ACTION.

The Office did not issue an advisory action. No other amendments have been filed subsequent to the FINAL ACTION.

(5)   Summary of Claimed Subject Matter

Broadly speaking, the claimed subject matter describes a technology for facilitating the automated generation of input-validation software filters. Using a user-interface for the described technology, a user is able to quickly enter a set of parameters (i.e., input-description-data) defining valid inputs. From this input-description-data, input-validation filters are automatically generated. An input-validation filter is a set of instructions for filtering input directly provided by a computing component without human intervention based upon the properties of valid input defined by the input-description-data.

Following is a concise explanation of each independent involved in the Appeal, including cites to the specification and specific reference characters. These specific reference characters are examples of particular elements of the drawings for certain claimed embodiments. It is to be appreciated and understood that the claims are not to be limited to solely the elements corresponding to these reference characters and that this section is provided to comply with the requirement of 37 CFR § 41.37(c)(1)(v).

Specifically:

Claim 1 includes obtaining input-description-data [p. 22, lines 3-5; block 210 of Fig. 2, p. 17, line 5 through p. 18, line 11; item 172 of Fig. 1]; transforming that input-description-data into a data structure [p. 22, lines 5-7; block 212 of Fig. 2, p. 18, line 13 through p. 19, line 9; item 176 of Fig. 1]; automatically generating a set of instructions for filtering input directly provided by a computing component without human intervention based upon the properties of valid input defined by the input-description-data. [p. 22, lines 9-15; block 216 of Fig. 2, p. 19,

line 11 through p. 20, line 2; item 180 of Fig. 1] The input-description-data define the properties of valid input directly provided by a computing component without human intervention and the data structure is an organized representation of the input-description-data.

Claim 19 includes obtaining input-description-data [p. 22, lines 3-5; block 210 of Fig. 2, p. 17, line 5 through p. 18, line 11; item 172 of Fig. 1]; transforming the input-description-data into a data structure [p. 22, lines 5-7; block 212 of Fig. 2, p. 18, line 13 through p. 19, line 9; item 176 of Fig. 1]; storing the data structure in a persistent form [p. 22, lines 5-8; block 214 of Fig. 2]; and automatically generating a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data. [p. 22, lines 9-15; block 216 of Fig. 2, p. 19, line 11 through p. 20, line 2; item 180 of Fig. 1] Here, the generating acquires the properties for generating the set of instructions from the data structure. The input-description-data define the properties of valid input provided by a computing component.

Claim 42 includes a user-interface for obtaining input-description-data [p. 22, lines 3-5; block 210 of Fig. 2, p. 17, line 5 through p. 18, line 11; item 172 of Fig. 1]; a transformer configured to transform the input-description-data into a data structure [p. 22, lines 5-7; block 212 of Fig. 2, p. 18, line 13 through p. 19, line 9; item 176 of Fig. 1]; a memory configured to store the data structure [p. 22, lines 5-8; block 214 of Fig. 2; item 906 of Fig. 3]; and a filter-instructions automatic generator ("autogen") configured to automatically generate a set of

instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data. [p. 22, lines 9-15; block 216 of Fig. 2, p. 19, line 11 through p. 20, line 2; item 180 of Fig. 1] Here, the filter-instructions autogen is further configured to acquire the properties from the data structure when automatically generating the set of instructions. The input-description-data define the properties of valid input provided by a computing component.

Claim 50 includes obtaining input-description-data [p. 22, lines 3-5; block 210 of Fig. 2, p. 17, line 5 through p. 18, line 11; item 172 of Fig. 1]; transforming the input-description-data into a data structure [p. 22, lines 5-7; block 212 of Fig. 2, p. 18, line 13 through p. 19, line 9; item 176 of Fig. 1]; storing the data structure in a persistent form [p. 22, lines 5-8; block 214 of Fig. 2]; and automatically generating a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data. [p. 22, lines 9-15; block 216 of Fig. 2, p. 19, line 11 through p. 20, line 2; item 180 of Fig. 1] Here, the generating acquires the properties for generating the set of instructions from the data structure. The input-description-data define the properties of valid input provided by a computing component.

Claim 54 includes obtaining input-description-data [p. 22, lines 3-5; block 210 of Fig. 2, p. 17, line 5 through p. 18, line 11; item 172 of Fig. 1] and automatically generating a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the

input-description-data. [p. 22, lines 9-15; block 216 of Fig. 2, p. 19, line 11 through p. 20, line 2; item 180 of Fig. 1] The input-description-data define the properties of valid input provided by a computing component.

(6)   Grounds of Rejection to be Reviewed on Appeal

A.   Whether 1-4, 6-20, 22-36, and 38-58 are obvious under USC § 103(a) and based upon the combination of **Fields** and **Lynch** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

B.   Whether claims 5 and 21 are obvious under USC § 103(a) and based upon the combination of **Fields, Lynch** and **Motoyama** disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com

lee&hayes

(7)   Argument

Issue A -- Whether 1-4, 6-20, 22-36, and 38-58 are obvious under USC § 103(a) and based upon the combination of Fields and Lynch disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

Cited References

The Office cites **Fields** as its primary reference and **Lynch** as it secondary reference in all of its obviousness-based rejections.

Fields

Fields describes a technology for automatically defining a filter used to extract web content for a web page, wherein the extracted content is used in a recast web page.

The recast web page may be produced by a hosting site, or may be part of an effort to revise a web site at a web content provider. First, a set of pages, possibly a single page, is retrieved from a content provider web server. Next, the web page is parsed to identify a set of selectable content elements. Next, a representation of the original web page is presented in a user interface, wherein the selectable content elements are demarcated. The user will select some of the elements for inclusion in the filter through the user interface, whereby the tool will indicate the selected content elements for inclusion in the filter.

Fields discloses the construction of the filter so that when the filter is used, the selected content elements are extracted from a retrieved web page from the content provider web server and reused in the recast web page. As part of the process of identifying the selectable content elements, a set of varied headers can be used to retrieve multiple versions of the same web page. In this way, the multiple versions of the web page are compared to identify static and dynamic content elements and marked as static or dynamic.

Lynch

Lynch describes an editor for allowing web authors to edit HTML visually while preserving the HTML source document.

The editor preserves the structure and format of the HTML, and permits simultaneous modeless visual and source document editing. When an edit is made with the invention, only the HTML source around that edit is updated, rather than rewriting the whole HTML source document.

Furthermore, when an edit is made, the new HTML source code is outputted in a format that is specified by the user. In order to preserve the format of the document, format information is stored in the parsed tree. The format of the node is preserved when its source is regenerated; edits to the node will reformat it according to user preferences. In order to preserve the structure of the document, invalid HTML structures are maintained and not corrected.

The editor will either support the invalid structure by reflecting such structure in the parsed tree (and thus allow for editing of the structure) or the editor will not support such a structure, and represent such structures as invalid

nodes. Moreover, the editor also maintains structure while editing, as the structure and format of the document is only minimally modified during editing, i.e. only the nodes affected by the edits are restructured and reformatted, and the remainder of the document is unmodified.

Claim 1

For the reader's convenience, the subject matter of this claim is provided below [with Office's cites to the references provided in brackets]:

obtaining input-description-data, **[Fields, col. 5, lines 15-25]** which define the properties of valid input directly provided by a computing component without human intervention; **[Lynch, col. 3, lines 30-60]**

transforming the input-description-data into a data structure, wherein the data structure is an organized representation of the input-description-data; **[Fields, col. 5, lines 15-25]**

from the organized representation of the input-description-data of the data structure, **[Fields, col. 5, lines 20-25]** automatically generating a set of instructions for filtering input directly provided by a computing component without human intervention **[Fields, col. 5, lines 1-30]** based upon the properties of valid input defined by the input-description-data. **[Lynch, col. 3, lines 30-60]**

Appellant respectfully submits that the Examiner failed to establish a *prima facie* case of obviousness for rejecting this claim in the NON-FINAL and FINAL ACTIONS. Applicant submits that the combination of **Fields** and **Lynch** does not disclose all of the elements and features of this rejected claim. Generally, the combination of the cited references does not disclose input-description-data being

transformed into a "data structure" which becomes the source of generated instructions and an automatic generation of a set of instructions.

More particularly, Applicant submits that neither reference discloses "from the organized representation of the input-description-data of the data structure, automatically generating a set of instructions...." In addition, Applicant submits that neither reference discloses the automatic generation of a set of instructions for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions.

FROM the data structure

Applicant submits that the combination of the cited references does not disclose a transformation of the "input-description-data" into a "data structure," which becomes the source of generated instructions. Rather, the references disclose an "HTML source" being transformed into an "HTML template," but that HTML template is not the source for generation of "filter definitions."

Applicant submits that **Fields** does not generate its "filter definitions" from the "HTML template." Rather, **Fields** generates its "filter definitions" from parsing of the "HTML source." In col. 9, lines 58-64, **Fields** discusses "filter definition" creation:

> The document filters can be created through several methods, including the analysis of the HTML source code, imbedded comments or delimiters and through comparisons with similar documents. Once the style of the web site is understood, a filter can be developed to look for the portion of the original document in which the hosting site is interested in reformatting.

Applicant submits that **Fields'** "filter definitions" are not produced by **Fields** from its "HTML template"; rather the definitions are produced by parsing its "HTML source." Therefore, **Fields** does not disclose what this claim recites.

The Office asserts that **Fields'** "HTML source" is equivalent to both the recited input ("input-description-data") and output ("the data structure") of the recited transformation. Applicant submits that the Office still has not identified where the cited references disclose such a transformation. In addition, the Office has not explained how the **Fields'** "HTML source" can be both the input and the output of a function. Therefore, neither of the cited references discloses what this claim recites.

Set of Instructions ≠ Filter Definition

Furthermore, Applicant submits that neither reference (**Fields** or **Lynch**) discloses the automatic generation of a "set of instructions" for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions. This claim recites the generation of a "set of instructions." Applicant submits that **Fields'** "filter definitions" are not the same as the recited "set of instructions."

At col. 12, line 48 through col. 22, line 24 and in U.S. Patent Application Serial No. 09/113,678, titled "Distribution Mechanism for Filtering, Formatting and Reuse of Web Based Content" (which is incorporated by reference into **Fields**), "filter definitions" are data and not a set of commands. Therefore, **Fields** supports this interpretation that "filter definitions" are not equivalent to "set of instructions." Applicant submits that **Fields'** "filter definitions" are not a "set of

instructions" as recited in the claims, rather the definitions are data and information.

Applicant submits that the Office has not explained how the Office can consider **Fields'** "filter definitions" to be the recited "set of instructions" when **Fields**, itself, indicates that its filter definitions include data instead of commands. Applicant respectfully submits that the Office has not shown that the combination of the cited references discloses all of the claimed features and elements.

Claims 2-18

These claims ultimately depend upon independent claim 1. As discussed above, claim 1 is allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

Claim 19

For the reader's convenience, the subject matter of this claim is provided below [with Office's cites to the references provided in brackets]:

> obtaining input-description-data, **[Fields, col. 5, lines 15-25]** which define the properties of valid input provided by a computing component; **[Lynch, col. 3, lines 30-60]**
> transforming the input-description-data into a data structure; **[Fields, col. 5, lines 15-25]**
> storing the data structures in a persistent form;
> automatically generating a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data, **[Fields, col. 5, lines 1-30]**

wherein the generating acquires the properties for generating the set of instructions from the data structure. [**Fields, col. 5, lines 20-25**]

Appellant respectfully submits that the Examiner failed to establish a *prima facie* case of obviousness for rejecting this claim in the NON-FINAL and FINAL ACTIONS. Applicant submits that the combination of **Fields** and **Lynch** does not disclose all of the elements and features of the rejected claims. Generally, the combination of the cited references does not disclose input-description-data being transformed into a "data structure" which becomes the source of generated instructions and an automatic generation of a set of instructions.

More particularly, Applicant submits that neither reference discloses: "wherein the generating acquires the properties for generating the set of instructions from the data structure." In addition, Applicant submits that neither reference discloses the automatic generation of a set of instructions for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions.

FROM the data structure

As discussed above on page 30, Applicant submits that the combination of the cited references does not disclose a transformation of the "input-description-data" into a "data structure," which becomes the source of generated instructions. Rather, the references disclose an "HTML source" being transformed into an "HTML template," but that HTML template is not the source for generation of "filter definitions."

The Office asserts that **Fields'** "HTML source" is equivalent to both the recited input ("input-description-data") and output ("the data structure") of the

recited transformation. Applicant submits that the Office still has not identified where the cited references disclose such a transformation. In addition, the Office has not explained how the **Fields'** "HTML source" can be both the input and the output of a function. Therefore, neither of the cited references discloses what this claim recites.

Set of Instructions ≠ Filter Definition

As discussed above on page 16, Applicant further submits that neither reference (**Fields** or **Lynch**) discloses the automatic generation of a "set of instructions" for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions. This claim recites the generation of a "set of instructions." Applicant submits that **Fields'** "filter definitions" are not the same as the recited "set of instructions."

Applicant submits that the Office has not explained how the Office can consider **Fields'** "filter definitions" to be the recited "set of instructions" when **Fields**, itself, indicates that its filter definitions include data instead of commands. Applicant respectfully submits that the Office has not shown that the combination of the cited references discloses all of the claimed features and elements.

Claims 20, 21, 24-33

These claims ultimately depend upon independent claim 19. As discussed above, claim 19 is allowable. As discussed above, claim 19 is allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

For the reader's convenience, the subject matter of this claim is provided below [with Office's cites to the references provided in brackets]:

> an user interface for obtaining input-description-data, **[Fields, col. 5, lines 15-25]** which define the properties of valid input provided by a computing component; **Lynch, col. 3, lines 30-60]**
> a transformer configured to transform the input-description-data into a data structure; **[Fields, col. 5, lines 15-25]**
> a memory, wherein the memory is configured to store the data structure;
> a filter-instructions automatic generator ("autogen") configured to automatically generate a set of instructions for filtering input provided by a computing component **[Fields, col. 5, lines 1-30]** based upon the properties of valid input defined by the input-description-data, wherein the filter-instructions autogen is further configured to acquire the properties from the data structure when automatically generating the set of instructions.**[Fields, col. 5, lines 20-25]**

Appellant respectfully submits that the Examiner failed to establish a *prima facie* case of obviousness for rejecting this claim in the NON-FINAL and FINAL ACTIONS. Applicant submits that the combination of **Fields** and **Lynch** does not disclose all of the elements and features of the rejected claims. Generally, the combination of the cited references does not disclose input-description-data being transformed into a "data structure" which becomes the source of generated instructions and an automatic generation of a set of instructions.

More particularly, Applicant submits that neither reference discloses: "wherein the filter-instructions autogen is further configured to acquire the properties from the data structure when automatically generating the set of

instructions." In addition, Applicant submits that neither reference discloses the automatic generation of a set of instructions for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions.

FROM the data structure

As discussed above on page 30, Applicant submits that the combination of the cited references does not disclose a transformation of the "input-description-data" into a "data structure," which becomes the source of generated instructions. Rather, the references disclose an "HTML source" being transformed into an "HTML template," but that HTML template is not the source for generation of "filter definitions."

The Office asserts that **Fields'** "HTML source" is equivalent to both the recited input ("input-description-data") and output ("the data structure") of the recited transformation. Applicant submits that the Office still has not identified where the cited references disclose such a transformation. In addition, the Office has not explained how the **Fields'** "HTML source" can be both the input and the output of a function. Therefore, neither of the cited references discloses what this claim recites.

Set of Instructions ≠ Filter Definition

As discussed above on page 16, Applicant further submits that neither reference (**Fields** or **Lynch**) discloses the automatic generation of a "set of instructions" for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions. This claim recites the generation

of a "set of instructions." Applicant submits that **Fields'** "filter definitions" are not the same as the recited "set of instructions."

Applicant submits that the Office has not explained how the Office can consider **Fields'** "filter definitions" to be the recited "set of instructions" when **Fields**, itself, indicates that its filter definitions include data instead of commands. Applicant respectfully submits that the Office has not shown that the combination of the cited references discloses all of the claimed features and elements.

Claims 46-49

These claims ultimately depend upon independent claim 42. As discussed above, claim 42 is allowable. As discussed above, claim 42 is allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

Claims 50

For the reader's convenience, the subject matter of this claim is provided below [with Office's cites to the references provided in brackets]:

> obtaining input-description-data, **[Fields, col. 5, lines 15-25]** which define the properties of valid input provided by a computing component; **[Lynch, col. 3, lines 30-60]**
> transforming the input-description-data into a data structure; **[Fields, col. 5, lines 15-25]**
> storing the data structures in a persistent form;
> automatically generating a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data, **[Fields, col. 5, lines 1-30]**

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com

lee☒hayes

wherein the generating acquires the properties for generating the set of instructions from the data structure. **[Fields, col. 5, lines 20-25]**

Appellant respectfully submits that the Examiner failed to establish a *prima facie* case of obviousness for rejecting this claim in the NON-FINAL and FINAL ACTIONS. Applicant submits that the combination of **Fields** and **Lynch** does not disclose all of the elements and features of the rejected claims. Generally, the combination of the cited references does not disclose input-description-data being transformed into a "data structure" which becomes the source of generated instructions and an automatic generation of a set of instructions.

More particularly, Applicant submits that neither reference discloses: "wherein the generating acquires the properties for generating the set of instructions from the data structure." In addition, Applicant submits that neither reference discloses the automatic generation of a set of instructions for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions.

FROM the data structure

As discussed above on page 30, Applicant submits that the combination of the cited references does not disclose a transformation of the "input-description-data" into a "data structure," which becomes the source of generated instructions. Rather, the references disclose an "HTML source" being transformed into an "HTML template," but that HTML template is not the source for generation of "filter definitions."

The Office asserts that **Fields'** "HTML source" is equivalent to both the recited input ("input-description-data") and output ("the data structure") of the

recited transformation. Applicant submits that the Office still has not identified where the cited references disclose such a transformation. In addition, the Office has not explained how the **Fields'** "HTML source" can be both the input and the output of a function. Therefore, neither of the cited references discloses what this claim recites.

Set of Instructions ≠ Filter Definition

As discussed above on page 16, Applicant further submits that neither reference (**Fields** or **Lynch**) discloses the automatic generation of a "set of instructions" for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions. This claim recites the generation of a "set of instructions." Applicant submits that **Fields'** "filter definitions" are not the same as the recited "set of instructions."

Applicant submits that the Office has not explained how the Office can consider **Fields'** "filter definitions" to be the recited "set of instructions" when **Fields**, itself, indicates that its filter definitions include data instead of commands. Applicant respectfully submits that the Office has not shown that the combination of the cited references discloses all of the claimed features and elements.

Claim 51

This claim ultimately depends upon independent claim 50. As discussed above, claim 50 is allowable. As discussed above, claim 50 is allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com

lee&hayes

Claim 54

For the reader's convenience, the subject matter of this claim is provided below [with Office's cites to the references provided in brackets]:

> obtaining input-description-data, **[Fields, col. 5, lines 15-25]** which define the properties of valid input provided by a computing component; **[Lynch, col. 3, lines 30-60]**
> automatically generating a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data. **[Fields, col. 5, lines 1-30]**

Appellant respectfully submits that the Examiner failed to establish a *prima facie* case of obviousness for rejecting this claim in the NON-FINAL and FINAL ACTIONS. Applicant submits that the combination of **Fields** and **Lynch** does not disclose all of the elements and features of the rejected claims. In particular, Applicant submits that neither reference discloses the automatic generation of a set of instructions for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions.

Set of Instructions ≠ Filter Definition

As discussed above on page 16, Applicant further submits that neither reference (**Fields** or **Lynch**) discloses the automatic generation of a "set of instructions" for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions. This claim recites the generation of a "set of instructions." Applicant submits that **Fields'** "filter definitions" are not the same as the recited "set of instructions."

Applicant submits that the Office has not explained how the Office can consider **Fields'** "filter definitions" to be the recited "set of instructions" when **Fields**, itself, indicates that its filter definitions include data instead of commands. Applicant respectfully submits that the Office has not shown that the combination of the cited references discloses all of the claimed features and elements.

Claims 55-58

These claims ultimately depend upon independent claim 54. As discussed above, claim 54 is allowable. As discussed above, claim 54 is allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

Issue B -- Whether claims 5 and 21 are obvious under USC § 103(a) and based upon the combination of Fields, Lynch and Motoyama disclosures and whether the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

These claims (5 and 21) ultimately depend upon independent claims 1 or 19. As discussed above, these independent claims are allowable. In addition to its own merits, each of these dependent claims is allowable for the same reasons that its base claim is allowable.

The Office relies on the combination of **Fields** and **Lynch** to reject the base independent claims 1 and 19. Applicant submits that the combination of **Fields** and **Lynch** does not disclose all of the elements and features of base independent claims 1 and 19 and, that **Motoyama** does not cure those deficiencies.

In particular, the combination of the cited references does not disclose input-description-data being transformed into a "data structure" which becomes the source of generated instructions and an automatic generation of a set of instructions.

For example, Applicant submits that neither reference discloses:

- "from the organized representation of the input-description-data of the data structure, automatically generating a set of instructions..." [claim 1];

- "wherein the generating acquires the properties for generating the set of instructions <u>from</u> the data structure." [claim 19]

In addition, Applicant submits that neither reference discloses the automatic generation of a set of instructions for filtering input. Instead, **Fields** discloses the automatic generation of "filter definitions," which are not instructions.

Accordingly, Applicant submits that claims 5 and 21 are allowable over the combination of **Fields, Lynch** and **Motoyama** for at least the reason that the references do not teach or suggest the combination of claimed elements and features.

Conclusion

Appellant respectfully submits that all of the Examiner's rejections have been traversed. As such, Appellant respectfully submits that all of the claims are in condition for allowance.

Respectfully Submitted,

Dated: 1.23 06

By: _____
Kasey C. Christie
Reg. No. 40559
(509) 324-9256 x232
kasey@leehayes.com
www.leehayes.com

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com

(8) Appendix of Appealed Claims

1. (PREVIOUSLY PRESENTED) A method for automatic production of one or more sets of instructions for an input filter of a computer system, the method comprising:

obtaining input-description-data, which define the properties of valid input directly provided by a computing component without human intervention;

transforming the input-description-data into a data structure, wherein the data structure is an organized representation of the input-description-data;

from the organized representation of the input-description-data of the data structure, automatically generating a set of instructions for filtering input directly provided by a computing component without human intervention based upon the properties of valid input defined by the input-description-data.

2. (ORIGINAL) A method as recited in claim 1, wherein the generating comprises translating the organized representation of the input-description-data of the data structure into the set of instructions.

3. (ORIGINAL) A method as recited in claim 2, wherein the translating comprises:

parsing the organized representation of the input-description-data of the data structure to acquire the input-description-data;

synthesizing the set of instructions based upon the input-description-data acquired by the parsing.

4.   (PREVIOUSLY PRESENTED)   A method as recited in claim 1 further comprising storing the data structure in a persistent form.

5.   (ORIGINAL)   A method as recited in claim 1, wherein the data structure is in a hierarchical markup language.

6.   (ORIGINAL)   A method as recited in claim 1, wherein the set of instructions as an input filter.

7.   (ORIGINAL)   A method as recited in claim 1 further comprising loading the set of instructions as an input filter.

8.   (ORIGINAL)   A method as recited in claim 1, wherein the set of instructions is generated with regard to filtering input for an application program module.

9.   (ORIGINAL)   A method as recited in claim 1, wherein input-description-data define the properties of input selected from a group consisting of valid input only, invalid input only, and both valid and invalid input.

10. (ORIGINAL)   A method as recited in claim 1, wherein the properties of valid input indicate parameters of input by defining boundary delimitations of such parameters and define assumptions regarding such parameters.

11. (ORIGINAL)   A method as recited in claim 1, wherein during the obtaining, input-description-data is obtained from a user via a graphical user interface.

12. (ORIGINAL)   A computer system comprising:

an application program module configured to receive and respond to input provided by a computing component;

an input filter module configured to receive input provided by a computing component for the application program module, filter the input, and pass the filtered input to the application program module,

wherein the filter comprises one or more sets of instructions that, when executed, filter the input and such sets of instructions being automatically produced according to the method as recited in claim 1.

13. (ORIGINAL)   A computer system as recited in claim 12, wherein the computer system comprises a Web server.

14. (ORIGINAL)   A computer system as recited in claim 12, wherein the input filter module is further configured to receive input from the computing component via a communications network.

15. (ORIGINAL)   A computer-readable medium comprising a set of instructions for filtering input, wherein such set of instructions has been automatically produced by the method as recited in claim 1.

16. (ORIGINAL)   An input filter of a computer having computer-executable instructions that, when executed, filter input, wherein such computer-executable instructions were automatically produced by the method as recited in claim 1.

17. (ORIGINAL)   A computer comprising one or more computer-readable media having computer-executable instructions that, when executed by the computer, perform the method as recited in claim 1.

18. (ORIGINAL)   A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 1.

19. (PREVIOUSLY PRESENTED) A method facilitating speedy and efficient production of one or more sets of instructions for an input filter of a computer system, the method comprising:

obtaining input-description-data, which define the properties of valid input provided by a computing component;

transforming the input-description-data into a data structure;

storing the data structures in a persistent form;

automatically generating a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data, wherein the generating acquires the properties for generating the set of instructions from the data structure.

20. (ORIGINAL) A method as recited in claim 19 further comprising transforming the input-description-data into a data structure

21. (ORIGINAL) A method as recited in claim 20, wherein the data structure is in a hierarchical markup language.

22. (CANCELED)

23. (CANCELED)

24. **(ORIGINAL)** A method as recited in claim 19 further comprising loading the set of instructions as an input filter.

25. **(ORIGINAL)** A method as recited in claim 19, wherein the properties of valid input indicate parameters of input by defining boundary delimitations of such parameters and define assumptions regarding such parameters.

26. **(ORIGINAL)** A method as recited in claim 19, wherein during the obtaining, input-description-data is obtained from a user via a graphical user interface.

27. **(ORIGINAL)** A computer-readable medium comprising a set of instructions for filtering input, wherein such set of instructions has been automatically produced by the method as recited in claim 19.

28. **(ORIGINAL)** An input filter of a computer having computer-executable instructions that, when executed, filter input, wherein such computer-executable instructions were automatically produced by the method as recited in claim 19.

29. (ORIGINAL) A computer system comprising:

an application program module configured to receive and respond to input provided by a computing component;

an input filter module configured to receive input provided by a computing component for the application program module, filter the input, and pass the filtered input to the application program module,

wherein the filter comprises one or more sets of instructions that, when executed, filter the input and such sets of instructions being automatically produced according to the method as recited in claim 19.

30. (ORIGINAL) A computer system as recited in claim 29, wherein the computer system comprises a Web server.

31. (ORIGINAL) A computer system as recited in claim 29, wherein the input filter module is further configured to receive input from the computing component via a communications network.

32. (ORIGINAL) A computer comprising one or more computer-readable media having computer-executable instructions that, when executed by the computer, perform the method as recited in claim 19.

33. (ORIGINAL) A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 19.

Claims 34-41 are **CANCELED**

42. (**PREVIOUSLY PRESENTED**)   An   automatic   filter-instructions production system comprising:

an user interface for obtaining input-description-data, which define the properties of valid input provided by a computing component;

a transformer configured to transform the input-description-data into a data structure;

a memory, wherein the memory is configured to store the data structure;

a filter-instructions automatic generator ("autogen") configured to automatically generate a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data, wherein the filter-instructions autogen is further configured to acquire the properties from the data structure when automatically generating the set of instructions.

43. (**CANCELED**)

44. (**CANCELED**)

45. (**CANCELED**)

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com

lee&hayes

46. (ORIGINAL)   A system as recited in claim 42, wherein the input-description-data indicate input parameters by defining boundary delimitations of such parameters and define assumptions regarding such parameters.

47. (ORIGINAL)   A computer-readable medium comprising a set of instructions for filtering input, wherein such set of instructions has been automatically produced by the system as recited in claim 42.

48. (ORIGINAL)   An input filter of a computer having computer-executable instructions that, when executed, filter input, wherein such computer-executable instructions were automatically produced by the system as recited in claim 42.

49. (ORIGINAL)   A system as recited in claim 42, wherein the interface is a graphical user interface.

50. (PREVIOUSLY PRESENTED)   A system for facilitating the production of one or more sets of instructions, the system comprising:

a memory comprising a set of computer program instructions; and

a processor coupled to the memory, the processor being configured to execute the computer program instructions, which comprise:

obtaining input-description-data, which define the properties of valid input provided by a computing component;

transforming the input-description-data into a data structure;

storing the data structures in a persistent form;

automatically generating a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data, wherein the generating acquires the properties for generating the set of instructions from the data structure.

51. (ORIGINAL) A system as recited in claim 50, wherein the input-description-data indicate input parameters by defining boundary delimitations of such parameters and define assumptions regarding such parameters.

52. (CANCELED)

53. (CANCELED)

54. (ORIGINAL) A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method comprising:

obtaining input-description-data, which define the properties of valid input provided by a computing component;

automatically generating a set of instructions for filtering input provided by a computing component based upon the properties of valid input defined by the input-description-data.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com

55.   (ORIGINAL)   A computer-readable medium as recited in claim 54, wherein the method further comprises loading the set of instructions as an input filter.

56.   (ORIGINAL)   A computer-readable medium as recited in claim 54, wherein the input-description-data indicate input parameters by defining boundary delimitations of such parameters and define assumptions regarding such parameters.

57.   (ORIGINAL)   An input filter comprising a computer-readable medium as recited in claim 54.

58.   (ORIGINAL)   A computer comprising one or more computer-readable media as recited in claim 54.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324.9256
F: 509.323.8979
www.leehayes.com

lee hayes